What is Claimed is:

1. A method comprising:

writing an entry to a store buffer;

determining said entry is in a first-level cache associated with said store buffer;

setting a status bit associated with said entry in said store buffer;

writing the entry to the first-level cache at retirement if the status bit is set; and

de-allocating the entry from said store buffer at retirement.

2. The method of claim 1 wherein writing an entry to a store buffer comprises:

writing an address and associated data for the entry to said store buffer.

3. The method of claim 2 wherein writing an address and associated data for the entry to said store buffer comprises:

writing the address for the entry to a store address buffer in said store buffer; and

writing an the data for the entry to a store data buffer in said store buffer.

4. The method of claim 1 wherein setting a status bit associated with said entry in said store buffer comprises:

writing a status bit to said store buffer from said first-level cache.

5. The method of claim 1 wherein writing the entry to the first-level cache at retirement when the status bit is set comprises:

determining the entry is being retired;

determining the status bit associated with the entry is set; and

writing the entry to the first-level cache.

6. The method of claim 1 wherein de-allocating the entry from said store buffer at retirement comprises:

re-setting the status bit associated with the entry.

7. The method of claim 6 further comprising:

re-setting the status bit associated with entries in the store buffer having addresses that correspond to the address of the entry.

8. The method of claim 1 further comprising:

re-setting the status bit associated with the entry, if the entry in the first-level cache is allocated over or snooped.

9. The method of claim 8 further comprising:

re-setting the status bit associated with entries in the store buffer having addresses that correspond to the address of the entry in the first-level cache, if the entry in the first-level cache is allocated over or snooped.

10. The method of claim 1 wherein determining said entry is in a first-level cache associated with said store buffer comprises:

sending a request to said first-level cache to determine whether said entry is in said first-level cache and is in a modified or exclusive state from said store buffer;

determining that said entry is in said first-level cache and is in a modified or exclusive state; and

receiving a write to set the status bit associated with the entry in said store buffer from said first-level cache.

11. The method of claim 1 wherein determining said entry is in a first-level cache associated with said store buffer comprises:

sending a request to said first-level cache to determine whether said entry is in said first-level cache and is in a modified or exclusive state from said store buffer;

determining that said entry is either not in said first-level cache or is in a modified or exclusive state;

reading in the entry from a next level cache; and

setting the status bit associated with each entry in said store buffer from that matches said entry.

12. The method of claim 1 wherein writing the entry to the first-level cache at retirement when the status bit is set comprises:

determining whether the status bit is set; and

writing the entry to the first-level cache from the store buffer, if the read for ownership bit is set.

13. The method of claim 1 wherein writing the entry to the first-level cache at retirement when the status bit is set comprises:

determining whether a read for ownership done bit is set;

reading the entry in to said first level cache from a next-higher level cache, if the status bit is not set; and

writing the entry to the first-level cache from the store buffer.

14. A machine-readable medium having stored thereon a plurality of executable instructions to perform a method comprising:

writing an entry to a store buffer;

determining said entry is in a first-level cache associated with said store buffer;

setting a status bit associated with said entry in said store buffer;

writing the entry to the first-level cache at retirement if the status bit is set; and

de-allocating the entry from said store buffer at retirement.

15. The machine-readable medium of claim 14 wherein writing an entry to a store buffer comprises:

writing an address and associated data for the entry to said store buffer.

16. The machine-readable medium of claim 15 wherein writing an address and associated data for the entry to said store buffer comprises:

writing the address for the entry to a store address buffer in said store buffer; and

writing the associated data for the entry to a store data buffer in said store buffer.

17. The machine-readable medium of claim 14 wherein setting a status bit associated with said entry in said store buffer comprises:

writing a status bit to said store buffer from said first-level cache.

18. The machine-readable medium of claim 14 wherein writing the entry to the first-level cache at retirement when the status bit is set comprises:

determining the entry is being retired;

determining the status bit associated with the entry is set; and

writing the entry to the first-level cache.

19. The machine-readable medium of claim 14 wherein de-allocating the entry from said store buffer at retirement comprises:

re-setting the status bit associated with the entry.

20. The machine-readable medium of claim 19 wherein the method further comprises:

re-setting the status bit associated with entries in the store buffer having addresses that correspond to the address of the entry.

21. The machine-readable medium of claim 14 wherein the method further comprises:

re-setting the status bit associated with the entry, if the entry in the first-level cache is allocated over or snooped.

22. The machine-readable medium of claim 21 wherein the method further comprises:

re-setting the status bit associated with entries in the store buffer having addresses that correspond to the address of the entry in the first-level cache, if the entry in the first-level cache is allocated over or snooped.

23. The machine-readable medium of claim 14 wherein determining said entry is in a first-level cache associated with said store buffer comprises:

sending a request to said first-level cache to determine whether said entry is in said first-level cache and is in a modified or exclusive state from said store buffer;

determining that said entry is in said first-level cache and is in a modified or exclusive state; and

receiving a write to set the status bit associated with the entry in said store buffer from said first-level cache.

24. The method of claim 14 wherein determining said entry is in a first-level cache associated with said store buffer comprises:

sending a request to said first-level cache to determine whether said entry is in said first-level cache and is in a modified or exclusive state from said store buffer;

determining that said entry is either not in said first-level cache or is in a modified or exclusive state;

reading in the entry from a next level cache; and

setting the status bit associated with each entry in said store buffer from that matches said entry.

25. The method of claim 14 wherein writing the entry to the first-level cache at retirement when the status bit is set comprises:

determining whether the status bit is set; and

writing the entry to the first-level cache from the store buffer, if the read for ownership bit is set.

26. The method of claim 14 wherein writing the entry to the first-level cache at retirement when the status bit is set comprises:

determining whether the status bit is set;

reading the entry in to said first level cache from a next-higher level cache, if the status bit is not set; and

writing the entry to the first-level cache from the store buffer.

27. A processor comprising:

a store buffer to store cache entry address and data information from a processor;

a request filter coupled to said store buffer; and

a first-level cache coupled to said request filter to store entry information;

said request filter to filter out duplicate requests for the entry from said store buffer.

28. The processor of claim 27 further comprising:

an execution unit coupled to said first-level cache and said store buffer.

29. The processor of claim 28 wherein said execution unit comprises:

a pipelined, out-of-order processor.

30. The processor of claim 28 wherein said store buffer comprises:

a store address buffer (SAB); and

a store data buffer (SDB).

31. The processor of claim 27 wherein said first-level cache comprises:

a write-back cache.

32. A processor comprising:

a front-end unit to fetch instructions, decode the instructions and store the decoded instructions;

an execution core coupled to said front-end unit to execute the decoded instructions;

a first-level cache coupled to said execution core to store data to be used to execute the decoded instructions;

a store buffer coupled to said execution core and said first-level cache to store new values resulting from the execution of the decoded instructions for the entries from said first-level cache; and

a retirement unit coupled to said execution component to receive results from the execution of said decoded instructions and to commit the results to the architectural state;

said processor including a machine-readable medium having stored thereon a plurality of executable instructions to perform a method comprising:

writing an entry to a store buffer;

determining said entry is in a first-level cache associated with said store buffer;

setting a status bit associated with said entry in said store buffer;

writing the entry to the first-level cache at retirement when the status bit is set; and

de-allocating the entry from said store buffer at retirement.

33. The processor of claim 32 wherein said store buffer comprises:

a store address buffer (SAB); and

a store data buffer (SDB).

34. The processor of claim 32 wherein said status bit comprises:

a read for ownership done bit.

35. A computer system comprising:

a memory; and

a processor coupled to the memory, the processor comprising:

a store buffer to store cache entry address and data information from a processor;

a request filter coupled to said store buffer; and

a first-level cache coupled to said request filter to store entry information;

said request filter to filter out duplicate requests for the entry from said store buffer.

36. The computer system of claim 35 wherein the processor further comprises:

an execution unit coupled to said first-level cache and said store buffer.

37. The computer system of claim 35 wherein the processor further comprises:

a pipelined, out-of-order processor.